NPS-MA-98-003

NAVAL POSTGRADUATE SCHOOL
Monterey, California

PARALLEL VERSION OF SPECIAL PERTURBATIONS ORBIT PROPAGATOR

by

Beny Neta

December 9, 1998

Prepared for: Naval Postgraduate School
Monterey, CA 93943

NAVAL POSTGRADUATE SCHOOL
Monterey, California 93943-5000

RADM Robert C. Chaplin
Superintendent

R. Elster
Provost

This report was prepared for Office of Naval Research and funded by Air Force Research Laboratory AFRL/VSSS.

This report was prepared by:

Beny Neta
Professor of Mathematics

Reviewed by:

Guillermo Owen, Chair
Department of Mathematics

Released by:

D. W. Netzer
Associate Provost and
Dean of Research

REPORT DOCUMENTATION PAGE

Form approved

OMB No 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 9, 1998	January 1, 1998 – December 1998

4. TITLE AND SUBTITLE
Parallel Version of Special Perturbations Orbit Propagator

5. FUNDING
MIPR # NAFRLP62800493

6. AUTHOR(S)
Beny Neta

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943

8. PERFORMING ORGANIZATION REPORT NUMBER
NPS-MA-98-003

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory
3550 Aberdeen SE
Kirtland AFB, NM 87117-5861

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
Approved for public release; distribution is unlimited.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words.)

Parallelization can be achieved by either control or domain decomposition. The latter was tried for analytic (by Phipps et al[19,20,21]) and semianalytic (by Wallace[26]) propagators. Neal and Coffey[14] discuss the domain decomposition for special perturbations. The control decomposition idea is inefficient for analytic propagators (Phipps[19]), because the computation time is too short. In this report we discuss a control decomposition approach to parallelize a numerical orbit propagator which may be more computationally intensive.

14. SUBJECT TERMS
orbit determination, domain decomposition, control decomposition

15. NUMBER OF PAGES
15

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	

PARALLEL VERSION OF SPECIAL PERTURBATIONS ORBIT PROPAGATOR

Beny Neta
Naval Postgraduate School
Department of Mathematics
Code MA/Nd
Monterey, CA 93943

December 9, 1998

Abstract

Parallelization can be achieved by either control or domain decomposition. The latter was tried for analytic (by Phipps et al[19,20,21]) and semianalytic (by Wallace[26]) propagators. Neal and Coffey[14] discuss the domain decomposition for special perturbations. The control decomposition idea is inefficient for analytic propagators (Phipps[19]), because the computation time is too short. In this report we discuss a control decomposition approach to parallelize a numerical orbit propagator which may be more computationally intensive.

Acknowledgements

This research was supported in part by Air Force Research Laboratory AFRL/VSSS at Kirtland AFB and by the Naval Postgraduate School Direct Funded Research Program.

1 Introduction

The orbit of an earth satellite may be predicted by solving the differential equations of Newtonian mechanics. The various approximate methods for solving the differential equation in this case may be divided into three categories: analytical, semianalytical, and numerical.

Within the next few years the military expects to increase the number of objects cataloged and to require more accurate predictions. With a commensurate improvement in sensor data, the accuracy of the predictions could be improved to the order of tens of meters (depending on time from epoch) with a semi-analytical theory or to meters with a completely numerical solution. However, these improvements would be obtained at the expense of a great increase in computation time using a numerical method on a serial computer.

Fonte et al[4] have compared the above three categories of propagators as implemented in R&D GTDS[9] in various orbital regimes. It was shown that accuracy of the Draper semianalytic (DSST) propagator when used with "optimal" parameters is close to that of the numerical, but the CPU time required is much less. Testing with real data is required to convince the users to adopt DSST. Also such a move requires training of potential users. Thus the most cost effective way to get more accurate predictions for more orbiting objects in a short amount of time would require the use of a parallel version of a numerical propagator.

In the next section we discuss parallel computing as applied to orbit propagation. In section 3, numerical propagators will be discussed. The experience gained with a control decomposition version of special perturbations orbit propagator is described in section 4.

2 Parallel Computing

Parallel computing is defined as the efficient form of information processing emphasizing the concurrent computations and manipulation of data to solve a single problem (see e.g. Hwang and Briggs[10]). Parallel computers may be classified according to their architecture. Flynn[3] has introduced a scheme to classify computers into four categories based on the multiplicity of instruction and data streams. The serial computers are called SISD (Single Instruction Single Data). Array processors are called SIMD, and most multiprocessors are MIMD. Another way of classification is by topology.

Parallel computing offers one option to decrease the computation time and achieve more real-time results. Use of parallel computers has already proven to be beneficial in reducing computation time in many other applied areas.

Two common measures of effectiveness, accounting for both the hardware and the algorithm are speedup and efficiency. The speedup, S_p, of an algorithm is defined as

$$S_p = \frac{T_s}{T_p} \qquad \text{or} \qquad \frac{T_1}{T_p} \tag{1}$$

where T_s is the time on a serial computer and T_i is the time on a parallel computer having i processors. The efficiency, E_p, is defined by

$$E_p = \frac{S_p}{p} \tag{2}$$

and it accounts for the relative cost of achieving a specific speedup. many factors could possibly limit the efficiency of a parallel program. These factors include the number of sequential operations that cannot be parallelized, the communication time between processors, and the time each processor is idle due to synchronization requirements, see e.g. Quinn[22].

2.1 Hypercube

The iPSC/2 INTEL hypercube[11] is a MIMD multicomputer with a hypercube topology. The computer consists of a system resource manager (host) and 2^n individual processors, called nodes (n is the dimension of the cube). The computing nodes may be augmented by a vector extension module for vector operations. Communication among the nodes and the host are completed through message passing. See Figure 1 for hypercubes with various dimensions n.

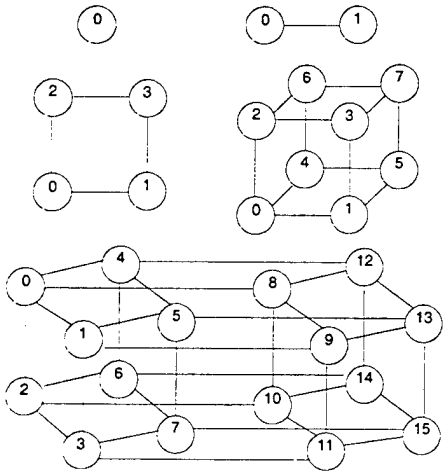

Figure 1: Hypercubes with dimension $n = 0, 1, 2, 3, 4$

2.2 Parallel Virtual Machines

Parallel Virtual Machine (PVM) is a small (~1 Mbytes of C source code) software package that allows a heterogeneous network of Unix-based computers to appear as a single large distributed-memory parallel computer. The PVM package is good for large-grain parallelism; that is, as least 100 kbytes/node. The term virtual machine is used to designate a logical distributed-memory computer and host is used to designate one of the member computers.

The PVM software, developed at Oak Ridge National Laboratory (see Dongara et al[2] and Sunderam et al[24]) supplied the functions to automatically start up tasks to communicate and synchronize with each other. A problem can be solved in parallel by sending and receiving messages to accomplish multiple tasks, similar to send and receive on the hypercube.

PVM handles all message conversion that may be required if two computers use different data representations. PVM also ensures that error messages generated on a remote computer are displayed on the user's local screen.

Parallelization could have been accomplished using a specific parallel multicomputer, such as the INTEL hepercube[11]. These systems tend to be large and expensive. While PVM may not accomplish the tasks as fast as, say, an INTEL iPSC/2 hypercube, the process execution times were satisfactory for the application tested.

2.3 Decomposition Strategies

Given a program and its associated data set, there are two primary ways to process it in parallel. The program can be separated into individual sections (called control decomposition) with a processor dedicated to compute its respective part, much like a factory assembly line. The other method domain decomposition is to divide up the data set and send parts to many separate processors all running the same algorithm, but on different data.

Figure 2 graphically presents these relationships between the node distributing data, the node collecting results and the workers.

In 1992, the first result on parallelization of orbit propagators was obtained by our student (W. E. Phipps[19,20,21]). These results were presented at the 1993 Space Surveillance Workshop at M.I.T. Lincoln Laboratory[20]. During the past five years a similar idea (domain decomposition) was applied to the analytic propagators SGP and SGP4/SDP4 (see Ostrom[18], Brewer[1], Neta et al[15], and Stone[23]). Our students developed a model to find the optimal number of processors (in the sense that the algorithm is most efficient). This optimal number depends on the satellite motion model used, the number of objects and the number of calls to the propagator for each object. Ostrom[18] and Neta et al[15] have shown that one can achieve near 100% efficiency. Wallace[26] suggested the same idea for the semianalytical propagator DSST. Neal and Coffey[14] demonstrated how to maintain the space catalog using similar parallelism idea for special perturbations.

The control decomposition idea for analytic propagators is inefficient (as demonstrated

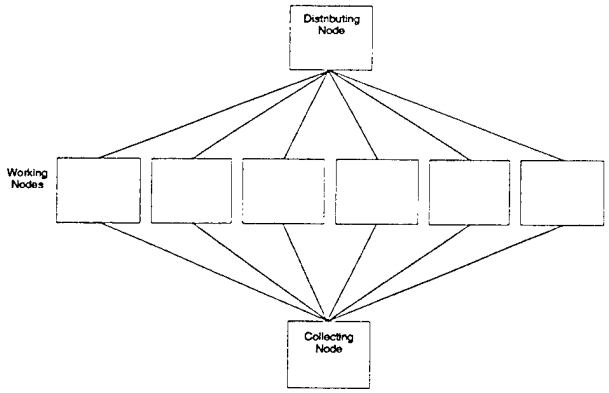

Figure 2: Domain decomposition

by Phipps[19]) since the analytic propagators are not computationally intensive.

In the next section, we discuss the possible numerical orbit propagators and their parallel version.

3 Numerical Propagators

Neta and Lustman have developed a parallel numerical ODE solvers for both linear[12] and nonlinear systems[13]. The idea here[13] is to use extrapolation. One can use Euler's or Gragg's method to solve the system

$$\frac{dy}{dt} = f(y(t), t), \qquad t_0 \leq t \leq b \tag{3}$$

subject to

$$y(t_0) = y_0 \tag{4}$$

For example, Gragg's method

$$\begin{aligned} z_{1/2} &= y_0 + \tfrac{h}{2} f(t_0, y_0) \\ y_1 &= y_0 + h f(t_{1/2}, z_{1/2}) \end{aligned} \tag{5a}$$

$$\begin{cases} z_{n+1/2} = z_{n-1/2} + h f(t_n, y_n) \\ y_{n+1} = y_n + h f(t_{n+1/2}, z_{n+1/2}) \end{cases} \qquad n = 1, 2, \cdots \tag{5b}$$

has truncation error

$$y_n - y(nh) = B_2h^2 + B_4h^4 + \cdots \tag{6}$$

Each one of the N processors uses an ODE solver with

$$h_r = \frac{N}{r}H, \tag{7}$$

thus the common points are $t_j = t_0 + (j-1)NH$.

Given

$$\{h_r, y(t_i, h_r) | r = 0, 1, \cdots, N-1; \quad i = 1, 2, \cdots, M\},$$

the solution at those M points in (t_0, b) is computed by the same scheme by all N possible $h's$). For polynomial extrapolation we construct a table of values T_{rs} as follows

$$\begin{aligned} T_{r0} &= y(t_i, h_r) \\ T_{rs} &= T_{r+1s-1} + \frac{T_{r+1s-1} - T_{rs-1}}{(\frac{h_r}{h_{r+s}})^2 - 1} \quad s = 0, 1, \cdots, N-1 \end{aligned} \tag{8}$$

Extrapolation will yield $O(h^{2N})$ accuracy for Gragg's scheme.

In the following table we show which processor computes which part of the solution.

	processor	step	proc 1	proc 2	proc 3	proc 4	proc 5	proc 6	proc 7
T_{00}	1	$8H$							
			T_{01}						
T_{10}	2	$4H$		T_{02}					
			T_{11}		T_{03}				
T_{20}	3	$\frac{8}{3}H$		T_{12}		T_{04}			
			T_{21}		T_{13}		T_{05}		
T_{30}	4	$2H$		T_{22}		T_{14}		T_{06}	
			T_{31}		T_{23}		T_{15}		T_{07}
T_{40}	5	$\frac{8}{5}H$		T_{32}		T_{24}		T_{16}	
			T_{41}		T_{33}		T_{25}		
T_{50}	6	$\frac{4}{3}H$		T_{42}		T_{34}			
			T_{51}		T_{43}				
T_{60}	7	$\frac{8}{7}H$		T_{52}					
			T_{61}						
T_{70}	8	H							

Table 1: Extrapolation assigned to each processor

The efficiency of this algorithm (based on low order integrator and extrapolation) is over 75%. These results were presented in Numerisk Institut in Denmark[16] and in the Fourth International Colloquium on Differential Equations in Bulgaria[17]. These algorithms were combined

with a numerical propagator to get a parallel special perturbations model. Fukushima[5] discusses the round-off error reduction in the extrapolation methods as they apply to orbital motion.

Fukushima[6] has suggested a numerical method based on Picard iteration and Chebyshev polynomials for approximating orbital motion. The idea is to integrate the differential equation

$$\frac{dy}{dt} = f(y(t), t), \qquad a \leq t \leq b \tag{9}$$

subject to

$$y(t_0) = y_0, \qquad a \leq t_0 \leq b. \tag{10}$$

Assuming $y^{(0)}(t)$ is an initial guess for the iteration, then

$$y^{(n+1)}(t) = y_0 + \int_{t_0}^{t} f(y^{(n)}(s), s)ds, \qquad a \leq t \leq b,\ n = 0, 1, \ldots \tag{11}$$

Both $y(t)$ and f are expanded in Chebyshev polynomials of the first kind, with coefficients Y_j, F_j, respectively.

Given all the coefficients $Y_j^{(n)}$, one can compute $y_k^{(n+1)} \equiv y^{(n)}(t_k^{(n+1)})$. This, in turn, will give the right hand side, $f_k^{(n+1)} \equiv f(y_k^{(n+1)}, t_k^{(n+1)})$. From this one can compute the coefficients $F_j^{(n+1)}$ and $Y_j^{(n+1)}$.

Fukushima[6] claims that "Clearly the present method is accelerated by using parallel computers. This is because the evaluation of the integrand can be done in parallel. Since the computational time of the numerical integration is mainly occupied by the integrand evaluation, we can expect a significant gain in real-time speed. In principle, the ratio of speed-up will become as many as the number of processors." This last sentence means that the efficiency is close to 100%. I believe that there is enough sequential work that will cause degradation of the efficiency, and one should experiment with the method on a parallel machine to get the actual efficiency. Several private communications with Fukushima reveal that in follow up papers[7,8], he developed a vectorization of the method on a Fujitsu VX/1R (with vector length of 2048). This vector method shows a gain of more than a 1000 times, which is around 50% efficiency.

For satellite problems, $y(t)$ and $f(y(t), t)$ in (9) are arrays.

$$y(t) = \begin{bmatrix} \vec{r} \\ \vec{v} \end{bmatrix} \qquad f(y(t), t) = \begin{bmatrix} \vec{v} \\ \vec{a} \end{bmatrix}$$

where $\vec{r}, \vec{v}$, and $\vec{a}$ are the position, velocity and acceleration vectors, respectively. Vallado[25] discusses fourth-order Runge-Kutta method (single step), as well as Runge-Kutta-Fehlberg (variable step size) and Adams-Bashforth-Moulton predictor-corrector (multi-step) methods. To obtain fourth-order or higher, one may use extrapolation with first-order Euler's method or second-order Gragg's method as discussed above. Using N processors and extrapolating Gragg's method, one can get $2N$-order accuracy. This idea is of control decomposition type. It yields higher efficiency than the Picard-Chebyshev method advocated by Fukushima.

4 SPEPH

In this subsection we give the results of our experimentation with SPEPH, the special perturbation code used by AFSPACECOM. We are **not** comparing the accuracies attained by various numerical propagators. We are only interested in the computation time in order to assess the feasibility of control decomposition. To this end, we have ran one example where the orbit is propagated to 15 minutes ahead. It is clear that the total run time of 3.19 seconds is too short to use control decomposition. If we now increase the length of period of propagation to 3 days and 15 minutes, then the total run time increases to 19.1 seconds, most of it (over 90%) is in the subroutine SPOOX, the SP integration driver. Even this is **not** computationally intensive enough. Several other orbits were propagated for various length of time and the total run time was always too short to justify control decomposition. I am sure that the reason that SPEPH is not used on all object is the fact that one requires many calls to the propagator to accomplish the differential corrections and update. Therefore one should consider the domain decomposition idea as implemented by Neal et al[14]. A more radical solution is to reconsider the differential correction process and see if one can save on the number of calls.

5 Conclusions

It has been shown that several methods in the literature yield an efficincy of over 75% for the solution of systems of first order ordinary differential equations such as the orbit prediction problem. We have access to a special perturbations code currently in use and our assessment of the efficiency of its control decomposition was discussed.

References

1. S. K. Brewer, Air Force Space Command Satellite Orbit Predictor Using Parallel Virtual Machines, M. Sc. Thesis, Naval Postgraduate School, Department of Mathematics, Monterey, CA, 1993.

2. J. Dongara, G. A. Geist, R. Mancheck, and V. S. Sunderam, Integrated PVM Framework Supports Heterogeneous Network Computing, Computers in Physics, pp 166-175, 1993.

3. M. J. Flynn, Very High - Speed Computing Systems, Proc. IEEE, **54**, (1966) 1901-1909.

4. D. J. Fonte, B. Neta, C. Sabol, D. A. Danielson and W. R. Dyar, Comparison of Orbit Propagators in the Research and Development Goddard Trajectory Determination System (R & D GTDS). Part I: Simulated Data, *Proc. AAS/AIAA Astrodynamics Conference*, Halifax, Nova Scotia, August 14-17, 1995, Paper Number 95-431.

5. T. Fukushima, Reduction of round-off errors in the extrapolation methods and its application to the integration of orbital motion, Astron. J., **112**, (1996), 1298-1301.

6. T. Fukushima, Picard iteration, Chebyshev approximation, parallel quadrature, and global numerical solution of perturbed dynamics, Astron. J., **113**, (1997).

7. T. Fukushima, Vector integration of dynamical motions by Picard-Chebyshev method, Astron. J., **113**, (1997).

8. T. Fukushima, Parallel/vector integration methods for dynamical astronomy, Cele. Mech. Dyn. Astr., Proc. IAU Coll. No. 172, Namur, July 1998.

9. Goddard Trajectory Determination System (GTDS) Mathematical Theory, NASA's Operational GTDS Mathematical Specification, Revision 1, Edited by Computer Science Corporation and NASA Goddard Space Flight Center, Contract NAS 5-31500, Task 213, July 1989.

10. K. Hwang, and F. A. Briggs, Computer Architecture and Parallel Processing, McGraw - Hill, New York, 1984.

11. iPSC/2 User's Guide, INTEL Corp., order number 311532-006, 1990.

12. L. Lustman, B. Neta, and C. P. Katti, Solution of linear systems of ordinary differential equations on an INTEL hypercube, SIAM J. Scientific and Statistical Computing, **12** (1991), 1480-1485.

13. L. Lustman, B. Neta, and W. Gragg, Solution of ordinary differential initial value problems on an INTEL hypercube, International J. Computers and Mathematics with Applications, **23** (1992), 65-72.

14. H. Neal, and S. Coffey, Maintaining the space catalog with special perturbations, *Proc. AAS/AIAA Astrodynamics Conference*, Sun Valley, Idaho, August 4-7, 1997, Paper Number 97-687.

15. B. Neta, D. A. Danielson, S. Ostrom, and S. K. Brewer, Performance of Analytic Orbit Propagator on a Hypercube and a Workstation Cluster, *Proc. AIAA-AAS Astrodynamics Program*, in Scottsdale, AZ, 1994, Paper Number 94-3706.

16. B. Neta, Parallel ODE Solvers, invited lecture at Numerisk Institut, Danmarks Tekniske Hojskole, Lyngby, Denmark, August 10, 1993.

17. B. Neta, Parallel Solution of Initial Value Problems, *Proc. Fourth International Colloquium on Differential Equations*, Plovdiv, Bulgaria, (D. Bainov, V. Covachev, A. Dishliev, eds.) August 18-23, **2** (1993), 19-42.

18. S. Ostrom, Parallelization of the Air Force Space Command (AFSPACECOM) Satellite Motion Models, M.S. thesis, Naval Postgraduate School, Department of Mathematics, Monterey, CA, 1993.

19. W. E. Phipps, Parallelization of the Navy Space Suveillance Center (NAVSPASUR) Satellite Motion Model, M.S. thesis, Naval Postgraduate School, Department of Mathematics, Monterey, CA, 1992.

20. W. E. Phipps, B. Neta and D. A. Danielson, Parallelization of the Navy Space Suveillance Center (NAVSPASUR) Satellite Motion Model, Journal Astronautical Sciences, **41**, (1993), 207-216.

21. W. E. Phipps, B. Neta and D. A. Danielson, Parallelization of the Navy Space Suveillance Center (NAVSPASUR) Satellite Motion Model, *Proc. 1993 Space Surveillance Workshop*, M.I.T. Lincoln Laboratory, Lexington, MA, March 30- April 1, 1993 (R. W. Miller and R. Sridharan, eds.), **2**, pp. 71-79.

22. M. J. Quinn, Designing Efficient Algorithms for Parallel Computers, McGraw-Hill, New York, 1987.

23. L. C. Stone, Parallel Processing of Navy Specific Applications Using a Workstation Cluster, MS Thesis, Naval Postgraduate School, Department of Electrical and Computer Engineering, Monterey, CA, December 1993.

24. V. S. Sunderam, G.A. Geist, and R. Mancheck, The PVM Concurrent Computing System: Evolution, Experience and Trends, J. Parallel Computing, **20**, (1994), 531-546.

25. D. A. Vallado, Fundamentals of Astrodynamics and Applications, McGraw Hill, 1997.

26. S. T. Wallace, Parallel Orbit Propagation and the Analysis of Satellite Constellations, Master of Sciences Thesis, Massachusetts Institute of Technology, Draper Laboratory Technical Report, CSDL-T-1245, Cambridge ,MA, June 1995.

Distribution List

	No. of copies
Director Defense Technical Information Center 8725 John J. Kingman Rd., Ste 0944 Ft. Belvoir, VA 22060-6218	2
Dean of Research Code 09 Naval Postgraduate School Monterey, CA 93943	1
Library Code 013 Naval Postgraduate School Monterey, CA 93943	2
Department of Mathematics Code MA Naval Postgraduate School Monterey, CA 93943	1
Professor D. A. Danielson Code MA/Dd Naval Postgraduate School Monterey, CA 93943	1
Professor Beny Neta Code MA/Nd Naval Postgraduate School Monterey, CA 93943	2

Professor Mike Ross
Code AA/Ro
Naval Postgraduate School
Monterey, CA 93943 — 1

Dr. Paul Cefola
The Charles Stark Draper Laboratory
555 Technology Square
Cambridge, Ma 02139 — 1

Dr. Ron Proulx
The Charles Stark Draper Laboratory
555 Technology Square
Cambridge, Ma 02139 — 1

Mr. Wayne McClain
The Charles Stark Draper Laboratory
555 Technology Square
Cambridge, Ma 02139 — 1

Lt. Chris Sagovac, USN
819 South Charles St.
Baltimore, MD 21230 — 1

Dr. Byron Tapley
Center for Space Research
University of Texas at Austin
Austin, TX 78712 — 1

Dr. David B. Spencer
AFRL/VSSS
3550 Aberdeen Ave SE
Kirtland AFB, NM 87117-5776 — 1

Lt Col Dave Vallado
AFRL/DEX
3550 Aberdeen Ave SE
Kirtland AFB, NM 87117-5776 — 1

Mr. Leo Early 1
2579A West 235^{th} St.
Torrance, CA 90505

Dr. Hunt Small 1
Lockheed-Martin Missiles and Space Co.
Dept E3-10
1111Lockheed Way
Sunnyvale, CA 94089-3504

Dr. Shannon Coffey 1
Code 8233
Naval Research Laboratory
Washington, DC 20375-5355

Dr. Joseph J. F. Liu 1
Schriever AFB, CO 80912-7202

Mr. Andrew E. Turner 1
3825 Fabian Way
Palo Alto, CA 94303

Dr. Chia-Chun (George) Chao 1
P. O. Box 92957-M4/946
Los Angeles, CA 90009-2957

Dr. Johnny H. Kwok 1
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena, CA 91109-8099

Col Sal Alfano 1
NSP/AN
250 S. Peterson Blvd 116
Peterson AFB, CO 80914-3180

Dr. Paul Schumacher
U.S. Naval Space Command
Code VN63T
Dahlgren, VA 22448-5170 1

James R. Wright
305 Exton Commons
Exton, PA 19341 1

George Rosborough
Campus Box 429
University of Colorado
Boulder, CO 80309 1

Mr. Richard Hujsak
Applied Technology Associates
305 Exton Commons
Exton, PA 19341 1

Dr. Felix R. Hoots
General Research Corporation
985 Space Center Drive
Suite 310
Colorado Springs, CO 80915 1